Proyecto de Diagnóstico de Cáncer de Mama

Cátedra de Inteligencia Artificial - 2020

Farber, Juan
—
Facultad de Ciencias Exactas y Tecnología - UNT
San Miguel de Tucumán, Tucumán

Flores Wittich, Pablo José
—
Facultad de Ciencias Exactas y Tecnología - UNT
San Miguel de Tucumán, Tucumán

Abstracto – Este documento contiene la resolución del proyecto final: "Diagnóstico de clase de Cáncer de Mama" planteado por la Cátedra de Inteligencia Artificial de la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán para el periodo lectivo 2020.

I. MARCO TEÓRICO GENERAL DE REDES NEURONALES

Las redes neuronales son modelos simples del funcionamiento del sistema nervioso. Las unidades básicas son las neuronas, que generalmente se organizan en capas, como se muestra en la siguiente ilustración.

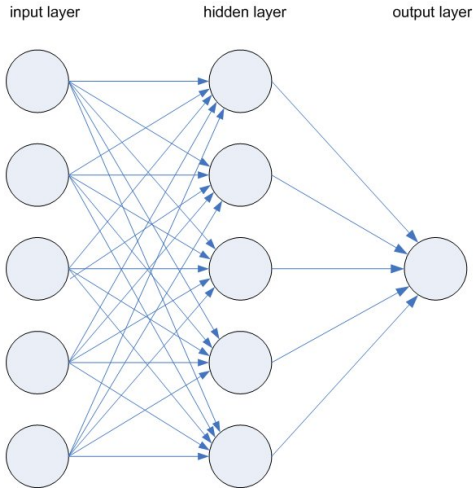

Figura 1: Ilustración de una red neuronal

Una red neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona ejecutando en simultáneo un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal: **una capa de entrada**, con unidades que representan los campos de entrada; una o varias **capas ocultas**; y una **capa de salida**, con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente. Al final, se envía un resultado desde la capa de salida.

La red aprende examinando los registros individuales, generando una predicción para cada registro y realizando ajustes a las ponderaciones cuando realiza una predicción incorrecta. Este proceso se repite muchas veces y la red sigue mejorando sus predicciones hasta haber alcanzado uno o varios criterios de parada.

Al principio, todas las ponderaciones son aleatorias y las respuestas que resultan de la red son, posiblemente, disparatadas. La red aprende a través del entrenamiento. Continuamente se presentan a la red ejemplos para los que se conoce el resultado, y las respuestas que proporciona se comparan con los resultados conocidos. La información procedente de esta comparación se pasa hacia atrás a través de la red, cambiando las ponderaciones gradualmente. A medida que progresa el entrenamiento, la red se va haciendo cada vez más precisa en la replicación de resultados conocidos. Una vez entrenada, la red se puede aplicar a casos futuros en los que se desconoce el resultado.

El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano. Los proyectos de redes neurales modernas suelen trabajar desde unos miles a unos pocos millones de unidades neuronales y millones de conexiones que, si bien son muchas órdenes, siguen siendo de una magnitud menos compleja que la del cerebro humano, más bien cercana a la potencia de cálculo de un gusano.

Las redes neuronales se han utilizado para resolver una amplia variedad de tareas, como la visión por computador y el reconocimiento de voz, que son difíciles de resolver usando la ordinaria programación basado en reglas.

Históricamente, el uso de modelos de redes neuronales marcó un cambio de dirección a finales de los años ochenta de alto nivel, que se caracteriza por sistemas expertos con conocimiento incorporado en las reglas si-entonces, a bajo nivel de aprendizaje automático, caracterizado por el

conocimiento incorporado en los parámetros de un modelo cognitivo con algún sistema dinámico.

Clasificación redes neuronales

Las redes pueden clasificarse según:

a. Topología de red

- Red neuronal Monocapa – Perceptrón simple

La red neuronal monocapa se corresponde con la red neuronal más simple, está compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos.

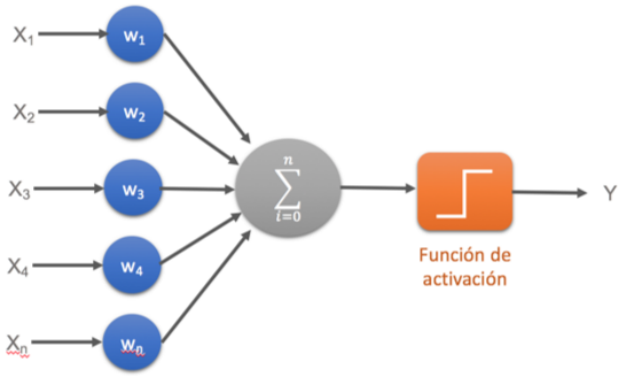

Figura 2: Perceptrón simple

- Red neuronal Multicapa – Perceptrón multicapa

La red neuronal multicapa es una generalización de la red neuronal monocapa, la diferencia reside en que mientras la red neuronal monocapa está compuesta por una capa de neuronas de entrada y una capa de neuronas de salida, está dispone de un conjunto de capas intermedias (capas ocultas) entre la capa de entrada y la de salida.

Dependiendo del número de conexiones que presente la red esta puede estar total o parcialmente conectada.

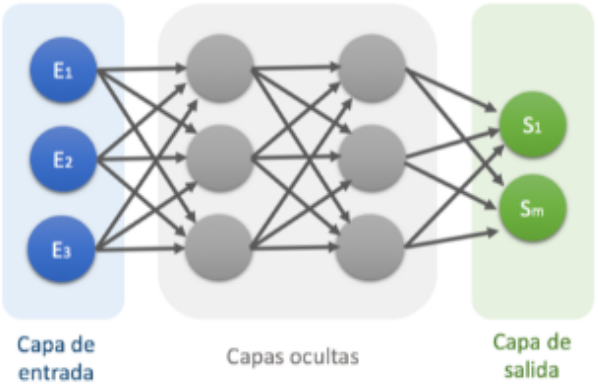

Figura 3: Perceptrón multicapa

- Red neuronal Convolucional (CNN)

La principal diferencia de la red neuronal convolucional con el perceptrón multicapa viene en que cada neurona no se une con todas y cada una de las capas siguientes sino que solo con un subgrupo de ellas (se especializa), con esto se consigue reducir el número de neuronas necesarias y la complejidad computacional necesaria para su ejecución.

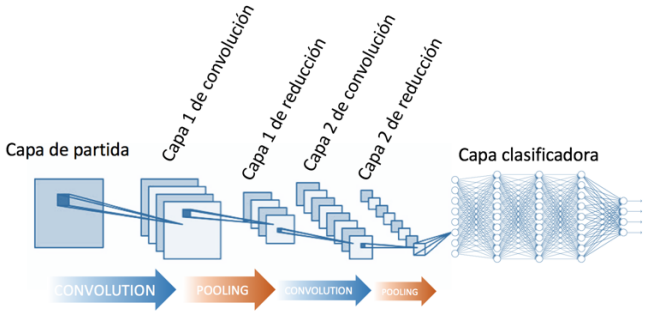

Figura 4: Red neuronal convolucional

- Red neuronal recurrente (RNN)

Las redes neuronales recurrentes no tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria.

Los datos introducidos en el momento t en la entrada, son transformados y van circulando por la red incluso en los instantes de tiempo siguientes t + 1, t + 2, …

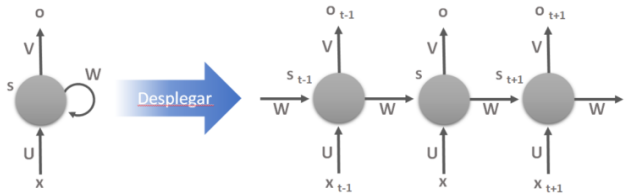

Figura 5: Red neuronal recurrente

- Redes de base radial (RBF)

Las redes de base radial calculan la salida de la función en función de la distancia a un punto denominado centro. La salida es una combinación lineal de las funciones de activación radiales utilizadas por las neuronas individuales.

Las redes de base radial tienen la ventaja de que no presentan mínimos locales donde la retropropagación pueda quedarse bloqueada.

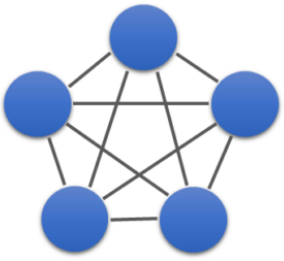

Figura 6: Red de base radial

b. Método de aprendizaje

Las redes neuronales son básicamente una imitación en software del funcionamiento de las neuronas de los animales: de sus interconexiones y de cómo ciertos estímulos de entradas producen ciertas salidas o resultados. Por otro lado, el aprendizaje automático consiste básicamente en dotar a los ordenadores de inteligencia artificial permitiéndoles aprender, y una forma de hacerlo es utilizando redes neuronales. Finalmente, el aprendizaje profundo es la unión de diversos algoritmos o fórmulas de aprendizaje automático para enseñar a las máquinas cuestiones abstractas más avanzadas, tales como reconocer caras, traducir textos o «entender» el lenguaje hablado.

- Aprendizaje supervisado

Se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un supervisor que determina la respuesta que que se debe generar para cada entrada.

El supervisor controla la salida y si esta no es correcta, modifica los pesos de las conexiones, con el fin de que la salida obtenida se aproxime a la deseada.

A su vez el aprendizaje supervisado se puede subdividir en:

- Aprendizaje por corrección de error.

Ajusta los pesos de las conexiones de la red en función del error cometido, es decir la diferencia entre los valores esperados y los obtenidos.

- Aprendizaje estocástico

Realiza cambios aleatorios sobre los pesos va calculando se la predicción va mejorando o empeorando con cada uno de los cambios, quedándose evidentemente con los cambios que mejoren los resultados.

Figura 7: Clasificación de aprendizaje supervisado

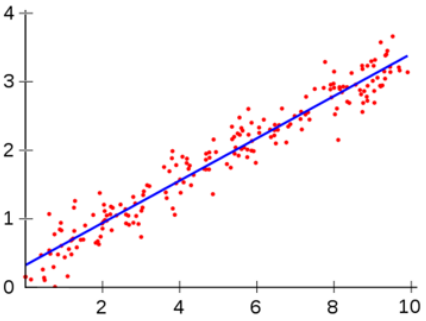

Figura 8: Regresión lineal.

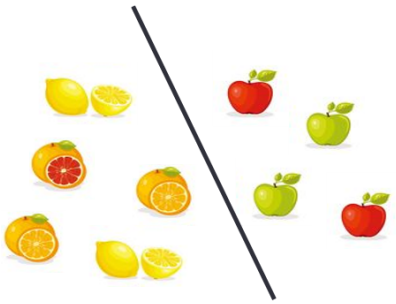

Figura 9: Clasificación.

- Aprendizaje no supervisado o autosupervisado

El aprendizaje supervisado son un conjunto de técnicas que permiten inferir modelos para extraer conocimiento de conjuntos de datos donde a priori se desconoce.

Las técnicas de aprendizaje no supervisado se pueden aplicar sin necesidad de tener los datos etiquetados para el entrenamiento.

Como sólo conocemos datos de entrada, pero no existen datos de salida que correspondan con las entradas, sólo se puede describir la estructura de los datos y con ello intentar encontrar algún tipo de organización que simplifique el análisis, por lo que tiene carácter exploratorio. No requieren influencia externa para ajustar los pesos.

Este tipo de aprendizaje busca encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten como entrada.

La interpretación de sus datos depende de su estructura y del algoritmo de aprendizaje empleado.

La salida podía representar el grado de similitud entre los datos, un clustering o establecimiento de categorías.

A su vez el aprendizaje no supervisado se puede subdividir en:

- Aprendizaje hebbiano

Permite medir la familiaridad o extraer las características de los datos de entrada.

- Aprendizaje competitivo y comparativo

Permite realizar clasificaciones de los datos de entrada.

La forma de actuación consiste en ir añadiendo elementos a una clase, si este nuevo elemento se determina que es de esta clase matiza los pesos, en caso contrario se puede crear una nueva clase con el elemento asociado a una serie de pesos propios.

Figura 10: Clasificación aprendizaje no supervisado.

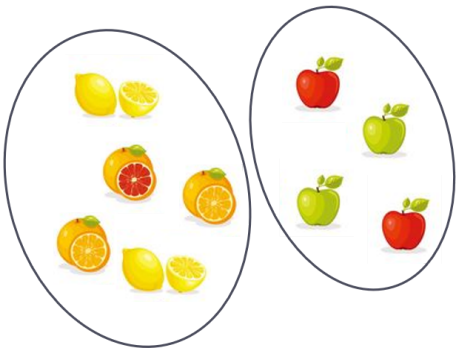

Figura 11: Análisis cluster.

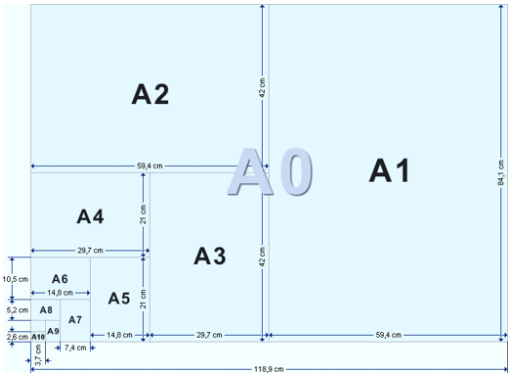

Figura 12: Reducción de dimensionalidad.

- Aprendizaje por refuerzo

Se considera un aprendizaje más lento que el aprendizaje por corrección de errores, en este caso no se dispone de un conjunto completo de los datos exactos de salida sino que se le indica solamente si el dato es aceptable o no, con esto el algoritmo ajusta los pesos basándose en un mecanismo de probabilidades.

II. Marco teórico particular del cáncer de mama

El cáncer es una enfermedad en la cual las células del cuerpo comienzan a multiplicarse sin control. Cuando el cáncer se origina en la mama (seno), se denomina cáncer de mama. Con la excepción del cáncer de piel, el cáncer de mama es el cáncer más común entre las mujeres. Muchas afecciones pueden causar bultos en las mamas, entre ellas el cáncer. Sin embargo, la mayoría de los bultos en las mama son causados por otras afecciones médicas. Las dos causas más comunes de bultos en las mama son la enfermedad fibroquística y los quistes. La enfermedad fibroquística causa cambios en la mama que no son cancerosos, le da una

consistencia grumosa y la hace sensible y dolorosa. Los quistes son pequeñas bolsas de líquido que pueden desarrollarse en las mama. Para diagnosticar los médicos suelen utilizar distintos tipos de pruebas para detectar o diagnosticar el cáncer de mama (seno), las cuales son:

- Ultrasonido mamario
- Mamografía diagnóstica
- Imágenes por resonancia magnética
- Biopsia

Una vez obtenido esos estudios, más los correspondientes análisis clínicos se realizarán pruebas para saber si las células cancerosas se han diseminado dentro de la mama o a otras partes del cuerpo. A este proceso se le llama clasificación por estadios o estadificación. El estadio del cáncer se determinará a partir de la localización del cáncer, es decir, si está únicamente en la mama (benigno) o si se ha extendido a los ganglios linfáticos de la axila o al resto del cuerpo (maligno). El médico podrá planificar el tratamiento de acuerdo al tipo de cáncer de mama y al estadio en que se encuentre.

III. Resolución problemática planteada

En el fichero wdbc.data tenemos los datos de 569 mujeres con cáncer de mama. Cada mujer está descrita por 32 atributos. El primero es un identificador, el segundo el tipo de cáncer (Maligno o Benigno) y el resto son el resultado de otros análisis clínicos. Pretendemos aprender el tipo de cáncer.

El objetivo es crear un sistema a partir de los datos cargados que nos permita indicar qué tipo de cáncer tiene la paciente para poder seguir el tratamiento correspondiente a cada uno, indicando el número de capas ocultas utilizadas; ya que no se brindó información del número de capas ni el número de neuronas que necesitan, también el número de neuronas de entrada y de salida y el algoritmo de entrenamiento a utilizar.

Dataset

Los 32 atributos que conforman al Dataset son:

1. ID: Número de identificación
2. Diagnóstico: El diagnóstico de los tejidos mamarios (M = maligno, B = benigno).
3. radius_mean: media de distancias desde el centro a puntos en el perímetro.
4. texture_mean: desviación estándar de los valores de escala de grises.
5. perimeter_mean: tamaño perimétrico del tumor del núcleo.
6. area_mean: tamaño promedio del área.
7. smoothness_mean: promedio de la suavidad.
8. radius_se: estandar para la media de las distancias desde el centro a los puntos en el perímetro
9. texture_se: estándar para la desviación estándar de los valores de escala de grises.
10. concavity_se: error estándar para la severidad de las porciones cóncavas del contorno.
11. concave points_se: error estándar para el número de porciones cóncavas del contorno.
12. symmetry_se: Simetría entre los puntos.
13. fractal_dimension_se: error estándar para "aproximación de la línea de costa" - 1.
14. radius_worst: "peor" o el mayor valor medio para la media de las distancias desde el centro hasta los puntos en el perímetro.
15. texture_worst: "peor" o el mayor valor medio para la desviación estándar de los valores de escala de grises.
16. perimeter_worst “peor” o el mayor valor medio del perímetro.
17. area_worst.: “peor” o el mayor valor medio del área.
18. smoothness_worst: "peor" o el mayor valor medio para la variación local en longitudes de radio
19. compactness_worst "peor" o el mayor valor medio para el perímetro ^ 2 / área - 1.0
20. concavity_worst: "peor" o el mayor valor medio para la gravedad de las porciones cóncavas del contorno
21. concave points_worst "peor" o el mayor valor medio para el número de porciones cóncavas del contorno
22. symmetry_worst.
23. fractal_dimension_worst "peor" o mayor valor medio para "aproximación de la línea de costa" - 1.

Preparación del Dataset para el análisis de datos

Por lo tanto, tenemos un total de 33 columnas, la última columna 'unnamed: 32' contiene todos los valores nulos, por lo que la excluiremos. Aparte de eso, tampoco incluiremos 'id' en nuestro conjunto de entrenamiento ya que no tiene ningún efecto en la clasificación. Por lo tanto, nos quedan 30 características que son del tipo float64 y no contienen valores faltantes. Ahora separemos las características y etiquetas.

Como el diagnóstico contiene 'M' o 'B' según sea maligno o benigno, debemos mapear este campo a 0 y 1 respectivamente.

Analisis y visualizacion de los datos

Mostramos el diagrama de cuentas generado para el dataset. Se clasifica la cantidad de diagnósticos vs el tipo de

diagnóstico, el cual puede ser “Maligno” o “Benigno”. Se obtienen los siguientes valores:

- Benignos = 357. Representa el 62.7416% del total.
- Malignos = 212. Representa el 37.2583% del total,

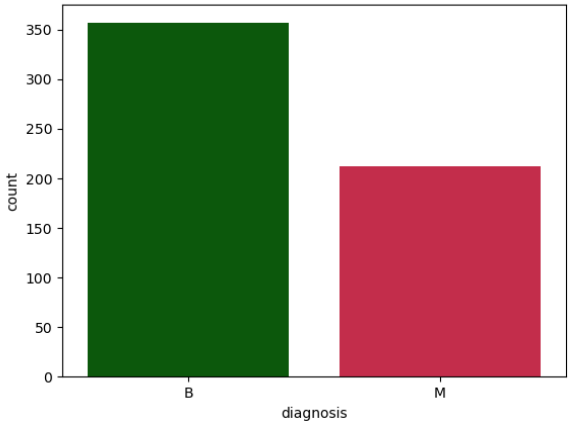

Figura 13: Diagrama de cuentas

Luego generamos un Heatmap para encontrar la correlación entre variables del dataset. Queremos ver la fuerza de la relación entre ellas.

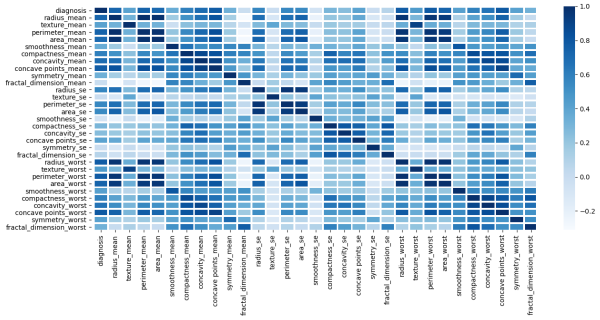

Figura 14: Heatmap.

Del Heatmap podemos observar la fuerza de las siguientes relaciones:

- **Alta:** radius_worst y perimeter_worst

La correlación entre ellas es aproximadamente 1.

- **Baja:** radius_mean y fractal_dimension_mean

La correlación entre ellas es aproximadamente -0.1.

A continuación se realizaron distintas mediciones en ambas relaciones para exponer sus diferencias, ya que son casos antagónicos.

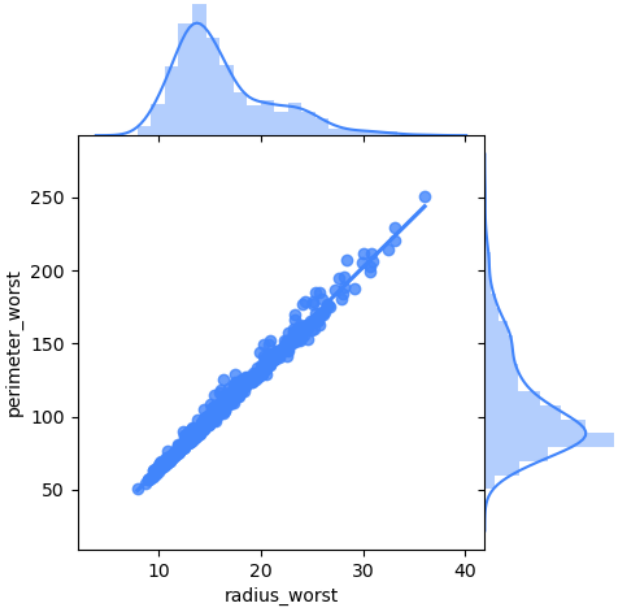

Figura 15: Regresión lineal de alta correlación.

En efecto, podemos ver donde se ubica la mayor parte de la muestra de cada variable. A su vez, graficamos la línea de tendencia. Este tipo de líneas puede decir si un conjunto de datos en particular han aumentado o decrementado en un determinado período

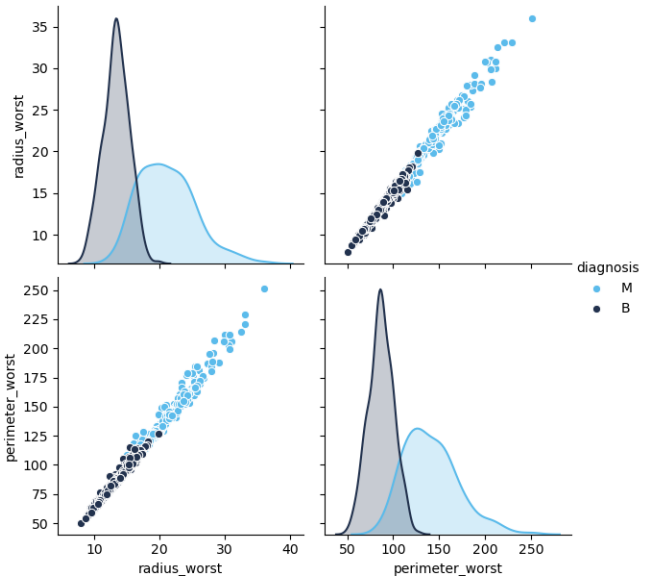

Figura 16: Diagrama de pares de alta correlación

Un diagrama de pares nos permite ver tanto la distribución de variables individuales como las relaciones entre dos variables. Podemos ver la tendencia y donde se ubica la mayor parte de nuestra muestra de datos, según el tipo de diagnóstico.

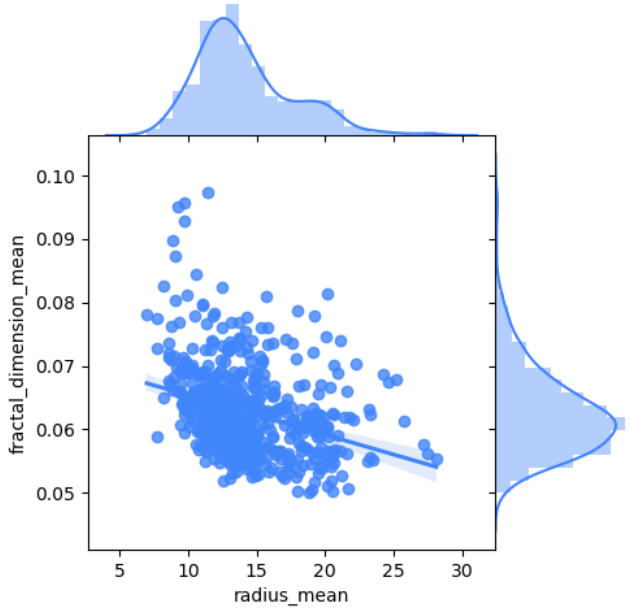

Figura 17: Regresión lineal de baja correlación.

Podemos ver que los puntos están muy dispersos, no es posible establecer una tendencia con el metodo de regresion lineal. No es acertado este análisis en particular.

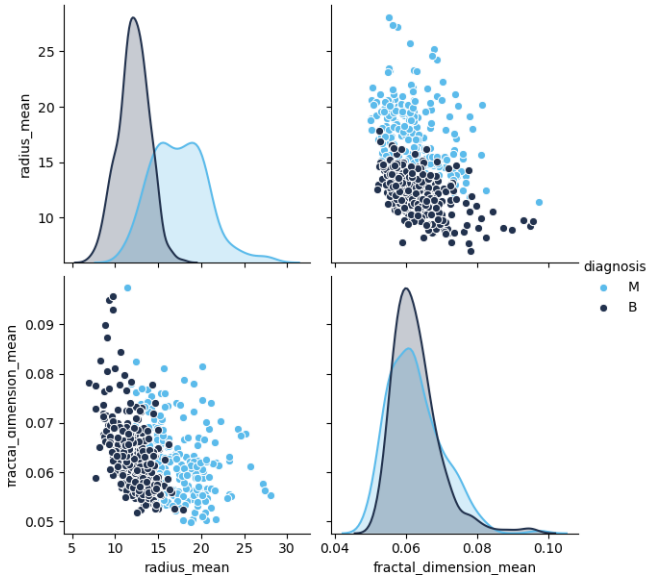

Figura 18:Diagrama de pares de baja correlación

Otra manera de visualizar, discriminado por tipo de diagnóstico.

Capas

Tendremos 4 capas, donde la topología es la siguiente:

Capa de entrada: 16 unidades con función de activación ReLU. Dimensión del input = 30.

Capa oculta 1: 8 unidades con función de activación ReLU

Capa oculta 2: 6 unidades con función de activación ReLU

Capa de salida: 1 unidad con función activación Sigmoidea

Utilizamos 2 funciones de activación:

ReLU o Rectified Lineal Unit: Transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran. Usamos ReLU para filtrar qué información se propaga a través de la red.

Sigmoid o Sigmoidea: La función sigmoide transforma los valores introducidos a una escala (0,1), donde los valores altos tienden de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0. La razón principal por la que usamos la función sigmoide es porque existe entre (0 a 1). Por lo tanto, se usa especialmente para modelos en los que tenemos que predecir la probabilidad como resultado. Dado que la probabilidad de que algo exista solo entre el rango de 0 y 1, esta función es la elección correcta para nuestra última capa.

- Buen rendimiento en la última capa.
- Lenta convergencia, puede hacer que una red neuronal se atasque en el momento del entrenamiento.

Separamos datos para entrenamiento y prueba

Los datos que usamos se dividen en datos de entrenamiento y datos de prueba. El conjunto de entrenamiento contiene una salida conocida y el modelo aprende sobre estos datos para ser generalizado a otros datos más adelante. Tenemos el conjunto de datos de prueba para probar la predicción de nuestro modelo.

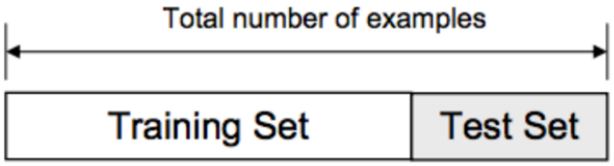

Figura 19: Datos para entrenamiento y pruebas

Utilizaremos los siguientes valores:

- **Entrenamiento:** El 80% de los datos.
- **Prueba:** El 20% de los datos.

Entrenamiento

Los hiper parámetros son las variables que determinan la estructura de la red (por ejemplo, número de unidades ocultas) y las variables que determinan cómo se entrena la red (por ejemplo, tasa de aprendizaje). Establecemos los siguientes parámetros:

- **Entrada**

epochs: Es el número de veces que se muestran todos los datos de entrenamiento a la red durante el mismo. Debemos aumentar este número hasta que la precisión de las pruebas comience a disminuir incluso cuando la precisión del entrenamiento aumente (problema del sobreajuste).

El sobreajuste se refiere a un modelo que modela los datos de entrenamiento demasiado bien. Esto ocurre cuando un modelo aprende el detalle, incluyendo el ruido en los datos de entrenamiento en la medida en que tiene un impacto negativo en el rendimiento del modelo en datos nuevos. Esto significa que el ruido o las fluctuaciones aleatorias en los datos de entrenamiento son recogidos y aprendidos por el modelo. El problema es que estos conceptos no se aplican a los datos nuevos y afectan negativamente a la capacidad de los modelos para generalizar

batch size: El tamaño del lote es el número de submuestras que se le dan a la red después de lo cual ocurre la actualización de parámetros. Hay que ir probando valores hasta que se encuentra el indicado

optimizer: Los optimizadores actualizan los parámetros de peso para minimizar la función de pérdida. La función de pérdida actúa como guía para el optimizador si se está moviendo en la dirección correcta para llegar al mínimo global. Utilizaremos la búsqueda de cuadrícula.

La búsqueda de cuadrícula, o Grid search, se utiliza para encontrar los hiper parámetros óptimos de un modelo que da como resultado las predicciones más "precisas".

Para ello, construiremos el clasificador, quien nos indicará los parámetros ideales para el entrenamiento de nuestro modelo. El clasificador compara los resultados de las siguientes dos funciones, y elegirá la los mejores resultados basándose en la precisión:

- Función Adam
- Función rms prop.

Compilación

Se encontraron los siguientes resultados:

```
best_parameters: {'batch_size': 1, 'epochs': 100, 'optimizer': 'rmsprop'}

best_accuracy: 0.978021978022
```

Figura 20:Búsqueda de hiper parámetros.

Según los hiper parámetros encontrados, la red se entrenará con cien iteraciones. El tamaño del lote será 1 y la función de optimización que mejor resultados dio fue rms prop (basándose en la precisión).

Guardamos los resultados obtenidos como un modelo para poder cargarlo posteriormente y no tener que rehacer la compilación (y por ende la búsqueda de los hiper parámetros) cada vez que se ejecuta el programa.

Clasificación

Asignaremos los resultados verdaderos o falsos en función de sus probabilidades (si la probabilidad >= 0.5 entonces es verdadera y sino falsa). Nuestra clasificación será la siguiente:

- **Output < 0.5:** Maligno (1).
- **Output >= 0.5:** Benigno (0).

Matriz de confusión

Es la métrica utilizada para encontrar y visualizar la precisión del modelo de clasificación.

		Predicted Values	
		Good	Bad
Actual Values	Good	True Positive (TP)	False Negative (FN)
	Bad	False Positive (FP)	True Negative (TN)

Figura 21:Explicando la matriz de confusión.

True positive: Caso donde el valor real y el valor predicho son verdaderos. El paciente ha sido diagnosticado con cáncer, y el modelo también predijo que el paciente tenía cáncer.

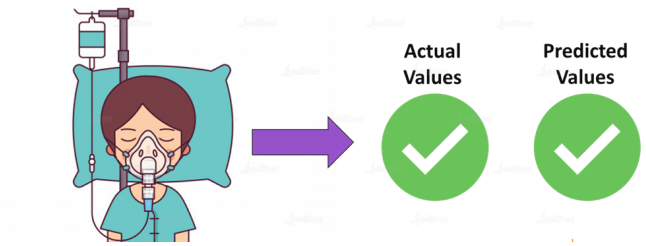

Figura 22:True positive.

False negative: Caso donde el valor real es verdadero, pero el valor predicho es falso, lo que significa que el paciente tiene cáncer, pero el modelo predijo que el paciente no tenía cáncer.

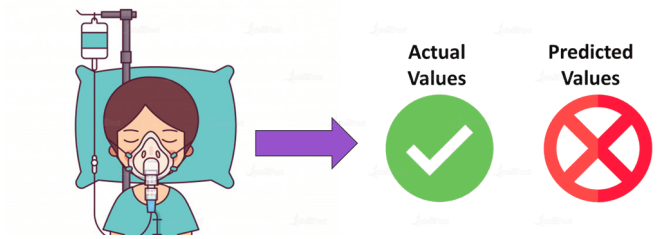

Figura 23:False negative.

False positive: Caso donde el valor predicho es verdadero, pero el valor real es falso. Aquí, el modelo predijo que el paciente tenía cáncer, pero en realidad, el paciente no tiene cáncer.

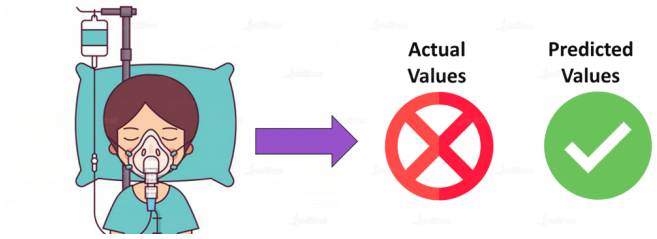

Figura 24:False positive

True negative: Caso donde el valor real es falso y el valor predicho también es falso. En otras palabras, al paciente no se le diagnostica cáncer y nuestro modelo predijo que el paciente no tenía cáncer.

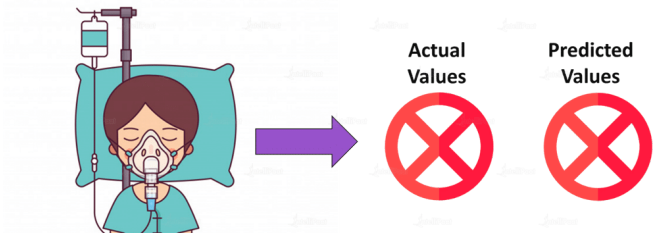

Figura 25:True negative

La matriz de confusión obtenida es la siguiente:

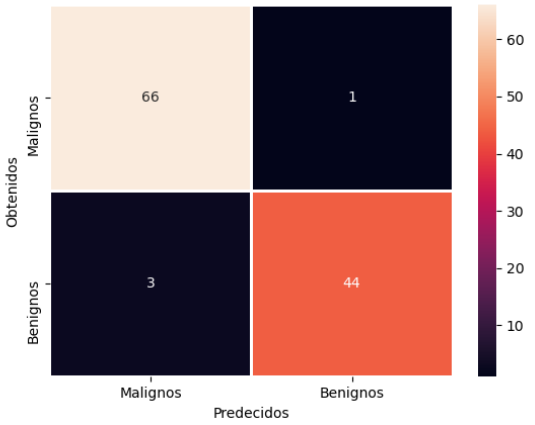

Figura 26:Matriz de confusión obtenida

En problemas de clasificación, "precisión" se refiere al número de predicciones correctas realizadas por el modelo predictivo sobre el resto de las predicciones. La precisión de nuestro modelo se calcula de la siguiente manera:

$$Precisión = \frac{Predicciones\ bien\ hechas}{Total\ de\ predicciones}$$

Por lo tanto, al particularizar la ecuación obtenemos lo siguiente:

$$Precisión = \frac{66 + 44}{66 + 1 + 3 + 44} = 0.96$$

Es decir, la precisión fue de un 96,49%.

IV. Elaboración de conclusiones

Para realizar este proyecto e introducirnos en el mundo de la inteligencia artificial, tuvimos que aprender a usar el lenguaje de programación Python y particularmente sus siguientes librerías:

Keras: Utilizada para encontrar los hiper parámetros y entrenar el modelo.

Matplotlib: Utilizada para la visualización de los datos.

TensorFlow: Necesaria para que Keras pueda operar.

Scikit-learn: Provee los algoritmos de aprendizaje para que Keras pueda operar y Seaborn o Matplotlib graficar.

Panda: Utilizada para poder manipular el Dataset.

Seaborn: Utilizada para la visualización de los datos.

Creemos que las habilidades aprendidas podrán ser muy provechosas en el futuro.

Por otro lado, estamos orgullosos de haber podido desarrollar un proyecto interdisciplinario y de gran relevancia actual como lo es la lucha contra el cáncer.

Con respecto a las redes neuronales, creemos que las grandes cantidades de datos disponibles recopilados durante la última década y el aumento del poder computacional han contribuido en gran medida a su popularidad. Esto ha permitido que las redes neuronales realmente muestren su potencial, ya que mejoran a medida que más datos ingresan y que su procesamiento este al alcance de todos, incluyendonos.

V. Bibliografía utilizada

[1] "AI with Python". Tutorials Point . 2016.

[2] Apunte de cátedra de Inteligencia Artificial, Facultad de Ciencias exactas y Tecnología, UNT.

VI. Sitios web utilizados

[1] https://kaggle.com/

[2] https://www.geeksforgeeks.org/

[3] https://towardsdatascience.com/

[4] https://intellipaa